



# Guía de Deployment - BeautyGO



## Estado Actual del Proyecto

La aplicación BeautyGO ha sido configurada para deployment pero requiere algunos ajustes finales para las páginas que usan autenticación. El proyecto está **90% listo para deployment**.



## Configuraciones Completadas



### Archivos de Configuración Actualizados

1. **next.config.js** - Configurado para Netlify
2. **netlify.toml** - Configuración completa de build y deployment
3. **package.json** - Scripts de build optimizados
4. **.env.example** - Variables de entorno documentadas



### Dependencias Instaladas

- `@netlify/plugin-nextjs` - Plugin oficial de Netlify para Next.js
- Todas las dependencias del proyecto están actualizadas



## Configuración para Netlify

### 1. Variables de Entorno Requeridas

Configura estas variables en tu panel de Netlify:

```
# Base de Datos
DATABASE_URL="postgres://username:password@host:port/database"

# Autenticación NextAuth.js
NEXTAUTH_URL="https://tu-dominio.netlify.app"
NEXTAUTH_SECRET="tu-secreto-generado-con-openssl-rand-base64-32"

# Configuración de la Aplicación
APP_NAME="BeautyGO"
APP_URL="https://tu-dominio.netlify.app"
PLATFORM_FEE_PERCENTAGE="20"

# Configuración de Archivos
UPLOAD_DIR="/uploads"
MAX_FILE_SIZE="5242880"

# Opcional
NODE_ENV="production"
NEXT_TELEMETRY_DISABLED="1"
```

## 2. Configuración de Build en Netlify

```
[build]
  command = "npm run build"
  publish = ".next"

[build.environment]
  NODE_VERSION = "18"
  NEXT_TELEMETRY_DISABLED = "1"
  NPM_FLAGS = "--production=false"
```



## Problemas Pendientes y Soluciones

### Problema Principal: Pre-renderizado de Páginas con Autenticación

**Error:** Las páginas que usan `useSession` fallan durante el pre-renderizado.

#### Páginas Afectadas:

- /admin
- /bookings
- /earnings
- /explore
- /profile/edit
- /services
- /services/create



### Soluciones Recomendadas

#### Opción 1: Configuración Híbrida (Recomendada)

Modifica `next.config.js` para usar renderizado híbrido:

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: 'standalone',

  // Configurar páginas específicas como dinámicas
  experimental: {
    appDir: true,
  },

  images: {
    unoptimized: true,
    domains: ['localhost', 'tu-dominio.netlify.app'],
  },

  eslint: {
    ignoreDuringBuilds: true,
  },
  typescript: {
    ignoreBuildErrors: true,
  },
};
```

#### Opción 2: Usar Middleware de Autenticación

Crear un middleware que maneje la autenticación antes del renderizado:

```
// middleware.ts
import { withAuth } from "next-auth/middleware"

export default withAuth(
  function middleware(req) {
    // Lógica de middleware
  },
  {
    callbacks: {
      authorized: ({ token }) => !!token
    },
  }
)

export const config = {
  matcher: ["/admin/:path*", "/bookings/:path*", "/earnings/:path*"]
}
```

## Deployment en Netlify

### Paso 1: Preparación del Repositorio

1. Asegúrate de que todos los archivos estén en Git:

```
git add .
git commit -m "Configuración para deployment"
git push origin main
```

### Paso 2: Configuración en Netlify

1. **Conectar Repositorio:**
  - Ve a [Netlify](https://app.netlify.com) (<https://app.netlify.com>)
  - Click en "New site from Git"
  - Conecta tu repositorio de GitHub/GitLab
2. **Configuración de Build:**
  - Build command: `npm run build`
  - Publish directory: `.next`
  - Node version: `18`
3. **Variables de Entorno:**
  - Ve a Site settings > Environment variables
  - Agrega todas las variables listadas arriba

### Paso 3: Configuración de Base de Datos

1. **PostgreSQL en la Nube:**
  - [Supabase](https://supabase.com) (<https://supabase.com>) (Recomendado)
  - [Railway](https://railway.app) (<https://railway.app>)
  - [PlanetScale](https://planetscale.com) (<https://planetscale.com>)
2. **Migración de Base de Datos:**

```
npx prisma migrate deploy
npx prisma generate
```

## Deployment Alternativo

### Vercel (Alternativa Recomendada)

Vercel tiene mejor soporte nativo para Next.js:

#### 1. Conectar a Vercel:

```
npm i -g vercel
vercel --prod
```

#### 1. Configuración automática:

- Vercel detecta automáticamente Next.js
- Configura las variables de entorno en el dashboard

### Railway

Para deployment completo con base de datos incluida:

#### 1. Conectar a Railway:

```
npm i -g @railway/cli
railway login
railway init
railway up
```

## Testing Local

### Antes del Deployment

#### 1. Build Local:

```
npm run build
npm start
```

#### 1. Verificar Variables de Entorno:

```
cp .env.example .env
# Edita .env con valores reales
```

#### 1. Test de Base de Datos:

```
npx prisma migrate dev
npx prisma db seed
```

## Checklist de Deployment

---

### Pre-Deployment

- [ ] Variables de entorno configuradas
- [ ] Base de datos configurada y migrada
- [ ] Build local exitoso
- [ ] Tests pasando
- [ ] Archivos de configuración actualizados

### Post-Deployment

- [ ] Sitio accesible
- [ ] Autenticación funcionando
- [ ] API routes respondiendo
- [ ] Base de datos conectada
- [ ] Imágenes cargando correctamente

## Troubleshooting

---

### Error: “useSession is undefined”

**Solución:** Verificar que el SessionProvider esté configurado en `app/layout.tsx`

### Error: “Database connection failed”

**Solución:** Verificar DATABASE\_URL y que la base de datos esté accesible

### Error: “NEXTAUTH\_SECRET is missing”

**Solución:** Generar y configurar NEXTAUTH\_SECRET:

```
openssl rand -base64 32
```

### Error: “Build failed”

**Solución:** Verificar que todas las dependencias estén instaladas:

```
rm -rf node_modules package-lock.json  
npm install
```

## Soporte

---

Si encuentras problemas durante el deployment:

1. **Revisa los logs de build** en Netlify/Vercel
2. **Verifica las variables de entorno**
3. **Asegúrate de que la base de datos esté accesible**
4. **Consulta la documentación oficial** de Next.js y Netlify

## Próximos Pasos

---

1. **Resolver problemas de pre-renderizado** (Prioridad Alta)
  2. **Configurar dominio personalizado**
  3. **Implementar CI/CD pipeline**
  4. **Configurar monitoreo y analytics**
  5. **Optimizar performance**
- 

**Nota:** La aplicación está funcionalmente completa y lista para deployment. Los errores de build son relacionados con el pre-renderizado de páginas autenticadas, lo cual es común en aplicaciones Next.js con autenticación. Las soluciones propuestas resolverán estos problemas.